Properties of the Least Squares Temporal Difference learning algorithm

Kamil Ciosek

Abstract

This paper presents four different ways of looking at the well-known Least Squares Temporal Differences (LSTD) algorithm for computing the value function of a Markov Reward Process, each of them leading to different insights: the operator-theory approach via the Galerkin method, the statistical approach via instrumental variables, the linear dynamical system view as well as the limit of the TD iteration. We also give a geometric view of the algorithm as an oblique projection. Furthermore, there is an extensive comparison of the optimization problem solved by LSTD as compared to Bellman Residual Minimization (BRM). Also, we look at the case where the matrix being inverted in the usual formulation of the algorithm is singular and show that taking the pseudo-inverse is the optimal thing to do then. We then review several schemes for the regularization of the LSTD solution. Moreover, we describe a failed attempt to derive an asymptotic estimate for the covariance of the computed value function, as well as describe one particular Bayesian scheme where such a covariance could be plugged in. We then proceed to treat the modification of LSTD for the case of episodic Markov Reward Processes.

1 Novel contributions of this paper

The main contribution of the present paper is the bringing together of the many perspectives that have been applied to thinking about LSTD. However, there is also novel material. In particular, we note the following: in section 6, we give the first correct derivation of LSTD via instrumental variables. In this respect, the paper [6], although seminal and certainly right is its core message, suffers from several serious formal shortcomings such as entirely ignoring the fact that the Bellman operator may take the value function out of the class representable by linear function approximators – compare equations in section 5.2 of [6], in addition to being unnecessarily complicated in its technical aspects. In contrast, our derivation is barely a page long and addresses the formal issues well. In section 7, we provide, as compared to [19], an additional way of decomposing LSTD into an oblique projection as well as the proof of a formal fact that makes the label 'oblique projection' right in the first place. In section 9, we do not introduce any new formulae but we believe that a direct comparison of the optimization problems shown in the provided table is the first where all the given ways of looking at the problem are present in one place. In section 10, the convergence analysis of the matrix series, necessary to formally compete the derivation, is new. The section 13 is new but we do accept the essence must have been informally realized for some time before. Our analysis of what happens when the matrix that LSTD inverts becomes singular, done in section 14 is entirely new. In section 15, our the idea to use the linear dynamical system view of LSTD for regularization is new in this context, although is has been strongly inspired by [5], which however solves a different problem. The attempted covariance derivation of section 16 is entirely new. The analysis of the episodic LSTD solution in terms of the feature matrix, the MRP transition matrix and the mean reward vector in section 18 is new. Finally, a fact about the assumption behind LSTD proved in section 19 is also entirely new.

2 Motivation

The main practical problem that the LSTD algorithm solves is such: we are given a feed of data from a stochastic system, consisting of a state description in terms of features and of rewards. The task is to construct an abstraction that maps from states to values of states, where the value is defined as the discounted sum of future rewards. For example, the system may describe a chess game, the features of state may describe what pieces the players have while the reward signal corresponds to wither winning or losing the game. The value signal will then correspond to the value of having each particular piece. Note that this is not a general constant but may depend on the way the individual players play the game, for example the values may be different for humans than for computer players.

Associated with our problem setting is the question whether the value function is interesting in its own right, or whether we only need it to adjust the future behaviour of some aspect of the environment we can control (i.e. in our chess example. make a move). We believe that there is large scope of systems (for instance expert systems) where the focus will be on gaining insight into the behaviour of the stochastic system, but the decisions about whether or how to act will still be made manually by human controllers, on the basis of the value-function information. These are the cases where algorithms like LSTD are the most directly applicable. On the other side of the spectrum, there will also of course be situations where the value function estimate is used as a tool to automatically generate the best action on the part of the agent – such systems may also use value-function estimation algorithms of the kind of LSTD if they are chosen to operate within the policy iteration framework.

3 Problem setting and notation

We are concerned with the problem of finding the value function of a finite-state Markov Reward Process (MRP). We only have access to linear features of states and to the obtained rewards. More formally, denote as P the transition matrix of the MRP. For each state s we have a feature row-vector ϕ_s. The feature design matrix Φ_D gives the features of all states of the MRP, row-wise. We know, since we have an MRP, that a random reward obtained while leaving a given state only depends on that state. We introduce the vector r_D, the i-th element of which contains mean reward obtained while leaving the state i. We use ξ to denote a left eigenvector of P with eigenvalue one. Note that if the chain has a stationary distribution, it will correspond to one such eigenvector, but we do not require it. Indeed we treat each recurrent class separately and give the current class the index c. We also introduce the matrix $\Xi_c = \text{diag}(\xi)$. We now define expectations of functions of the Markov process in terms of weighted averages. For example the expectation of $\boldsymbol{\phi}^\top \boldsymbol{\phi}$, is defined by $\mathrm{E}_c\left[\boldsymbol{\phi}^\top \boldsymbol{\phi}\right] = \Phi_D^\top \Xi_c \Phi_D$, and similarly for other functions. Here, we mean that for each recurrent class c there exists such Ξ_c that it is legitimate to consider the above quantity an expectation corresponding to long-time average by the standard ergodic theorem for Markov chains (applied to states within the recursive class c). Note that we still allow for aperiodicity, i.e. the diagonal of Ξ_c may not be a stationary distribution, but the expression still matches the long-time average. We use subscripts do denote two-step sampling, for example $\boldsymbol{\phi}'_{\boldsymbol{s}}$ denotes the fact that we first sample a state, then the successor state and obtain the feature of that successor state. When we write an expectation w.r.t. such a variable, for example $\mathrm{E}_c\left[\boldsymbol{r}^2_{\boldsymbol{s}}\right]$, the distribution we mean for $\boldsymbol{r}$ is $\sum_{s=1}^{S} p(\boldsymbol{r}|s)\xi_s$. We note that the MRP is fixed, i.e. we only consider the on-policy setting. We use bold letters to denote random variables, for instance $\boldsymbol{s}$ denotes state and $\boldsymbol{\phi}$ denotes feature. Once we have obtained samples from our process, we store them in matrices Φ_S and r_S, whose i-th rows correspond to, respectively, the state feature vector and reward obtained at time i. Observe the difference between Φ_D and Φ_S – in the first one, each state is represented once, in the second one the number of rows corresponds to the trajectory taken in the MRP and repetitions are possible. The value function is discounted with the factor $0 < \gamma < 1$. Moreover, we introduce the square matrix D which has ones on the main diagonal and $-\gamma$ on the diagonal above it. It is the sample based equivalent to the operator $I - \gamma P$.

4 The approximation that LSTD makes

Now we can write the Bellman equation, which defines the true value function: $V(\boldsymbol{s}) = \mathrm{E}_c[\boldsymbol{r} + \gamma V(\boldsymbol{s}') \,|\, \boldsymbol{s}] = \mathrm{E}_c[\boldsymbol{r} \,|\, \boldsymbol{s}] + \gamma \,\mathrm{E}_c[V(\boldsymbol{s}') \,|\, \boldsymbol{s}]$. It can be rewritten in matrix form as $V_D = (I - \gamma P)^{-1} r_D$. Here, the vector V_D contains the value of the function $V(\cdot)$ at each state. Now the scope for the LSTD algorithm is where it is not possible to use this equation to compute the value function exactly (i.e. compute the table-lookup solution) for two reasons: (1) we may not have access to the states directly, just to functions ϕ_s and (2) even if we did, this would require sample-based versions of the matrices P, and r_D, the first of which being particularly intractable because the number of entries is square in the (already large) number of states. Thus we exploit a linear architecture: i.e. we seek to approximate the true value function $V(\cdot)$ with the function $\bar{V}(s) = \phi_s w$, which is linear in w; or, in the language of random variables $\bar{V}(\boldsymbol{s}) = \boldsymbol{\phi} w$. We will briefly discuss two possibilities for how to choose an appropriate $\bar{V}()$ within the linear class of functions. The obvious thing to do would be to define $\bar{V}_D = \Pi V_D = \Pi (I - \gamma P)^{-1} r_D$, where $\Pi = \Phi_D (\Phi_D^\top \Xi_c \Phi_D)^{-1} \Phi_D^\top \Xi_c$ where we note that the inverse is well-defined by assumption (A.1). This formula guarantees that the distance from $\bar{V}_D$ to V_D is minimal in the L^2 norm weighted by Ξ_c. The problem with this approach is that it is not known how to compute a useful estimate of the projected value function from samples [1]. Therefore we need a different approximation. We call it $\tilde{V}(\cdot)$. It comes through the equation $\tilde{V}_D = \Pi T \tilde{V}_D$, where we look for the fixpoint of the operator ΠT instead of the Bellman operator T, where $Tx = r_D + \gamma P \Phi_D x$. This is motivated further in the next section (see equation 3).

Now the question, of course, is about the relation between our approximation $\tilde{V}_D$ and the projection of the true value function $\bar{V}_D$, as we have defined it in the previous section. We now state without proof the relation between the two estimates developed in [25] (see their references for prior work).

$$V_D - \tilde{V}_D = (I - \gamma \Pi P)^{-1} (V_D - \bar{V}_D) \tag{1}$$

This can be used to obtain the following bound, which does not require us to estimate the matrices Π or P (see [25] for proof and for sharper bounds): $\|V_D - \tilde{V}_D\|_{\Xi_c} \leq (1 - \gamma^2)^{-1/2} \|V_D - \bar{V}_D\|_{\Xi_c}$. We see from this that one example where the approximation of equation 6 is appropriate is when we have substantial discounting – in that case, if the linear framework is good at all, i.e. if the projection $\bar{V}_D = \Pi V_D$ is close to the true value function, then so will be our approximation. We note that this bound concerns only one recursive class, which, i.e. the one corresponding to Π. It tells us nothing about the other classes (i.e. using this bound only, we have to accept the value function at the states belonging to them may be arbitrarily off the mark).

[1] One algorithm that can do that in the limit of infinitely many samples is Least-Squares Monte-Carlo. It is, however, prone to high variance in the estimate for small sample sizes.

In the subsequent sections, we will describe various seemingly different approaches to computing $\tilde{V}(\cdot)$ from samples, which however all lead to the same formula for the solution. In order to derive our algorithm, we make two assumptions. First, we assume that the matrix $\mathrm{E}_c\left[\boldsymbol{\phi}^\top \boldsymbol{\phi}\right]$ is full rank (A.1). This is often understood to mean that the features are linearly independent. This reflects a useful intuitive concept, but is not exactly true. We stress that the statement concerns *both* the features and the transition dynamics of the MRP, and means that the parts of the features corresponding to states visited with nonzero probability are independent. We note that this implies that the matrix $\mathrm{E}_c\left[\boldsymbol{\phi}^\top (\boldsymbol{\phi} - \gamma \boldsymbol{\phi}')\right]$ is also full rank (F.1) – we discuss why this implication holds in section 19. We will also use this to claim the invertability of $\Phi_S^\top D \Phi_S$ without further comment (i.e. we assume we have enough samples). Also, we assume that the mean of the reward process exists $\mathrm{E}_c[\boldsymbol{r}_{\boldsymbol{s}}] < \infty$ (A.2).

To summarize the description, we restate the fundamental conditions for LSTD to yield good value estimates: (1) the linear architecture itself needs to match the problem and the set of features needs to be set right, that is V_D must be close to $\bar{V}_D$, (2) the approximation $\tilde{V}_D$ needs to be good, for example through discounting and finally (3) the sample based approximation $\hat{w}$ to w must also be good (in the following sections we define an consistent estimator for w, i.e. a way to compute $\hat{w}$, so that the value function computed from a sample trajectory approaches $\tilde{V}_D$ for the recursive states in the class corresponding to that trajectory as the length of the trajectory goes to infinity).

5 Derivation using the Galerkin method

That LSTD corresponds to a special case of the Galerkin argument has been implicitly realized for some time, and formally stated in [3], on which this section is based. The general idea of the Galerkin method is to approximate the fixed point of T, $Tx^\star = x^\star$. We have $x^\star = \mathrm{argmin}_x \|Tx^\star - x\|$. We introduce the approximation by considering points from within the column space of Φ, so that our approximate fixpoint satisfies $\tilde{x}^\star \in \mathcal{C}(\Phi)$, yielding $\tilde{x}^\star = \mathrm{argmin}_{x \in \mathcal{C}(\Phi)} \|T\tilde{x}^\star - x\|$, which is equivalent to the following, after substituting $\Phi y^\star$ for $\tilde{x}^\star$ and Φy for x.

$$\Phi y^\star = \operatorname*{argmin}_y \|T\Phi y^\star - \Phi y\| \tag{2}$$

Now, for the L^2 weighted semi-norm with the corresponding projection operator Π, this has an analytic solution: $\Phi y^\star = \Pi(T(\Phi y^\star))$. Now, in our case $\Phi = \Phi_D$, $\Pi = \Phi_D(\Phi_D^\top \Xi_c \Phi_D)^{-1}\Phi_D^\top \Xi_c$ where we note that the inverse is well-defined by assumption (A.1), the evaluation of the operator T at Φw becomes $r_D + \gamma P \Phi_D w$ with w assuming the role of $y^\star$ and the norm $\|\cdot\|$ becomes the weighted norm $\|\cdot\|_{\Xi_c}$. Now we solve the following.

$$\Phi_D w = \Pi(\underbrace{r_D + \gamma P \Phi_D w}_{T\Phi_D w}) \tag{3}$$

This can be transformed in the following way.

$$\cancel{\Phi_D}(I - \gamma(\Phi_D^\top \Xi_c \Phi_D)^{-1}\Phi_D^\top \Xi_c P \Phi_D))w = \cancel{\Phi_D}(\Phi_D^\top \Xi_c \Phi_D)^{-1}\Phi_D^\top \Xi_c r_D \tag{4}$$

In the above, we can cancel out the terms Φ_D, because by assumption (A.1), Φ_D has to be of full column rank. We then multiply both sides by $(\Phi_D^\top \Xi_c \Phi_D)$, to obtain $\left((\Phi_D^\top \Xi_c \Phi_D) - \gamma \Phi_D^\top \Xi_c P \Phi_D\right) w = \Phi_D^\top \Xi_c r_D$, which leads to the following.

$$w = \left(\Phi_D^\top \Xi_c \Phi_D - \gamma \Phi_D^\top \Xi_c P \Phi_D\right)^{-1} \Phi_D^\top \Xi_c r_D \tag{5}$$

This is the same as the expression we will obtain in the instrumental variable section.

6 Derivation through instrumental variables

We will first obtain a statistical model that expresses the properties of the approximation $\tilde{V}(\cdot)$. By solving the Bellman equation directly in the linear approximation regime, we obtain the following equation.

$$\boldsymbol{\phi} w = \tilde{V}(\boldsymbol{s}) = \mathrm{E}_c[\boldsymbol{r} \,|\, \boldsymbol{s}] + \gamma \,\mathrm{E}_c\left[\tilde{V}(\boldsymbol{s}') \,\middle|\, \boldsymbol{s}\right] + e_{\boldsymbol{s}} = \mathrm{E}_c[\boldsymbol{r} \,|\, \boldsymbol{s}] + \gamma \,\mathrm{E}_c[\boldsymbol{\phi}' \,|\, \boldsymbol{s}]\, w + e_{\boldsymbol{s}} \tag{6}$$

We note that in the above, we introduce the convention that w is a column vector while the features are row vectors. This convention minimizes the number of transposes we have to write. Note that we had to introduce the TD error vector $e = [e_{s_1}, \cdots, e_{s_n}]^\top = T\Phi_D w - \Phi_D w$ and the corresponding random variable $e_{\boldsymbol{s}}$ (i.e. the error is a deterministic function of the current state, which is random), since the reward vector r_D and the expected feature vector $\mathrm{E}_c[\boldsymbol{\phi}' \,|\, \boldsymbol{s}]\, w$ may not be in the feature space (i.e. the column space of Φ_D). It can be verified using the result for w from the previous section that, the error terms satisfy $e\Xi_c\Phi_D = 0$, i.e. it is orthogonal to the feature space (F.2) (indeed it can be seen after a

brief manipulation that the condition $e\Xi_c\Phi_D = 0$ is equivalent to the formula 5), and that consequently $\Pi e = 0$. This is not a derivation from first principles, since we had to use an external argument to verify that $e\Xi_c\Phi_D = 0$. But given the model of equation 6 it is nonetheless instructive to look at the mechanics of how the derivation works because this is the first one to have been proposed for LSTD.

We now accept equation 6 as a given and give a statistical derivation as provided in the original LSTD paper [6], based on methods described in [24]. Now, because we do not observe the expectations $\mathrm{E}_c[\gamma V(\boldsymbol{s}') \,|\, \boldsymbol{s}]$ and $\mathrm{E}_c[\boldsymbol{r} \,|\, \boldsymbol{s}]$ in equation 6, but merely samples of $\boldsymbol{\phi}$ and $\boldsymbol{\phi}'$ we model the residue wrt. the expected value as noise, yielding the probabilistic model $\boldsymbol{r_s} = \mathrm{E}_c[\boldsymbol{r} \,|\, \boldsymbol{s}] + \boldsymbol{\eta_s}$, where we use assumption (A.2), and $\boldsymbol{\phi'_s} = \mathrm{E}_c[\boldsymbol{\phi}' \,|\, \boldsymbol{s}] + \boldsymbol{\varepsilon_s}$. Note that by definition $\mathrm{E}_c[\boldsymbol{\eta_s} \,|\, \boldsymbol{s}] = 0$. Observe that this implies that $\mathrm{E}_c[\boldsymbol{\eta_s} \,|\, \boldsymbol{\phi}] = \mathrm{E}_c[\mathrm{E}_c[\boldsymbol{\eta_s} \,|\, \boldsymbol{s}] \,|\, \boldsymbol{\phi}] = 0$ (F.3) by the law of iterated expectation (LIE). Analogously, $\mathrm{E}_c[\boldsymbol{\varepsilon_s} \,|\, \boldsymbol{\phi}] = 0$ (F.4).

Thus we can rewrite equation 6 to obtain the following.

$$\boldsymbol{\phi} w = \boldsymbol{r_s} + \gamma \boldsymbol{\phi'_s} w - \gamma \boldsymbol{\varepsilon_s} w - \boldsymbol{\eta_s} + e_{\boldsymbol{s}} \quad \text{or} \quad \boldsymbol{r_s} = (\boldsymbol{\phi} - \gamma \boldsymbol{\phi'_s}) w + \underbrace{\gamma \boldsymbol{\varepsilon_s} w + \boldsymbol{\eta_s}}_{\boldsymbol{\zeta_s}} - e_{\boldsymbol{s}} \tag{7}$$

Now, we cannot use traditional least-squares to solve this, since the expression $\boldsymbol{\zeta_s} = \gamma \boldsymbol{\varepsilon_s} w + \boldsymbol{\eta_s}$ may be, in general, correlated[2] with $\boldsymbol{\phi} - \gamma \boldsymbol{\phi'_s}$, so will be the projection error term e and the two will not cancel in general. Therefore the noise term $\boldsymbol{\zeta_s} - e_{\boldsymbol{s}}$ may be correlated with with $\boldsymbol{\phi} - \gamma \boldsymbol{\phi'_s}$. Also, $\mathrm{E}_c[e_s \,|\, s]$ is not necessarily zero. But ordinary least squares (OLS) requires that noise be uncorrelated with input variables and that it have mean zero to yield consistent estimates. However, there is still a way to obtain a good estimate. More formally, we first need to establish the following properties. First, we have $\mathrm{E}_c\big[\boldsymbol{\phi}^\top \boldsymbol{\eta_s}\big] = \mathrm{E}_c\big[\mathrm{E}_c\big[\boldsymbol{\phi}^\top \boldsymbol{\eta_s} \,\big|\, \boldsymbol{\phi}\big]\big] = \mathrm{E}_c\big[\boldsymbol{\phi}^\top \mathrm{E}_c[\boldsymbol{\eta_s} \,|\, \boldsymbol{\phi}]\big] = 0$, where the first equality follows from LIE and the second from fact (F.3). By the same reasoning, we have $\mathrm{E}_c\big[\boldsymbol{\phi}^\top \boldsymbol{\varepsilon_s}\big] = 0$ from fact (F.4). With these two properties, we can now multiply both sides of equation 7 by $\boldsymbol{\phi}^\top$, which we for this purpose call an *instrumental variable*, and then take expectation, so as to make the noise terms vanish. We also have $\mathrm{E}_c\big[\boldsymbol{\phi}^\top e_{\boldsymbol{s}}\big] = 0$ by fact (F.2). This results in the following.

$$\mathrm{E}_c\big[\boldsymbol{\phi}^\top \boldsymbol{r_s}\big] = \mathrm{E}_c\big[\boldsymbol{\phi}^\top (\boldsymbol{\phi} - \gamma \boldsymbol{\phi'_s})\big]\, w + \underbrace{\gamma\, \mathrm{E}_c\big[\boldsymbol{\phi}^\top \boldsymbol{\varepsilon_s}\big]\, w + \mathrm{E}_c\big[\boldsymbol{\phi}^\top \boldsymbol{\eta_s}\big] - \mathrm{E}_c\big[\boldsymbol{\phi}^\top e_{\boldsymbol{s}}\big]}_{=0} \tag{8}$$

Now because we know from fact (F.1) that $\mathrm{E}_c\big[\boldsymbol{\phi}^\top (\boldsymbol{\phi} - \gamma \boldsymbol{\phi'_s})\big]$ is invertible, the estimator w is given by the following.

$$w = \mathrm{E}_c\big[\boldsymbol{\phi}^\top (\boldsymbol{\phi} - \gamma \boldsymbol{\phi'_s})\big]^{-1} \mathrm{E}_c\big[\boldsymbol{\phi}^\top \boldsymbol{r_s}\big] = \big(\Phi_D^\top \Xi_c (I - \gamma P) \Phi_D\big)^{-1} \Phi_D^\top \Xi_c r_D \quad \text{or} \quad \hat{w} = (\Phi_S^\top D \Phi_S)^{-1} \Phi_S^\top r_S \tag{9}$$

This finishes the formal derivation. We will now give two different intuitive interpretations to the instrumental variable method. First, consider the sample equivalent of equation 7, which we now rewrite in matrix notation $r_S = D\Phi_S \hat{w} + \boldsymbol{\zeta}_S - e_S$, where by $\boldsymbol{\zeta}_S$ we denote the vector containing the noise terms for each individual sample and by e_S the sample values of the random varaible $e_{\boldsymbol{s}}$. Now, as described above, we cannot solve it by OLS because of the correlation between the noise and $D\Phi_S$. So we 'fix' $D\Phi_S$ by projecting it onto the feature space (i.e. the column space of Φ_S), since we know that noise is uncorrelated with features. We introduce the projection operator $\Pi_S = \Phi_S (\Phi_S^\top \Phi_S)^{-1} \Phi_S^\top$, where we note that the inverse exists by assumption (A.1) given we have enough samples. Now our equation becomes the following.

$$\Pi_S r_S = \Pi_S D \Phi_S \hat{w} + \underbrace{\Pi_S \boldsymbol{\zeta}_S}_{\to 0 \text{ as } N \to \infty} - \Pi_S e_S \quad \text{or} \quad \cancel{\Phi_S (\Phi_S^\top \Phi_S)^{-1}} \Phi_S^\top r_S = \cancel{\Phi_S (\Phi_S^\top \Phi_S)^{-1}} \Phi_S^\top D \Phi_S \hat{w} \tag{10}$$

In the above, we can cancel the terms because Φ_S has, by assumption (A.1), independent columns if we have enough samples. This leads to the same estimator that we derived above. This interpretation is known in econometric literature as two-stage least squares (2SLS), because we solve two linear systems: first we project $D\Phi_S$ on the subspace of features and then we solve the resulting modified equation. In this context we stress that we would get the same solution if we only applied the projection on the right-hand side, e.g. $r_S = \Pi_S D \Phi_S \hat{w}$ – this can be seen by noticing that the choice of $\hat{w}$ in this equation is unaffected by any component of r_S orthogonal to the feature space. We also see the direct correspondence between this and the projection step in the derivation through Galerkin method – the equation 3 is essentially the limiting version of the sample-based equation 10.

[2] Indeed, we have $\mathrm{E}_c\big[\boldsymbol{\phi'_s}^\top \boldsymbol{\eta_s}\big] = 0$, $\mathrm{E}_c\big[\boldsymbol{\phi}^\top \boldsymbol{\varepsilon_s}\big] = 0$ and $\mathrm{E}_c\big[\boldsymbol{\phi}^\top \boldsymbol{\eta_s}\big] = 0$ as shown later in the text; but $\mathrm{E}_c\big[\boldsymbol{\phi'_s}^\top \boldsymbol{\varepsilon_s}\big] = \mathrm{E}_c\big[\boldsymbol{\phi'_s}^\top \boldsymbol{\phi'_s}\big] - \mathrm{E}_c\big[\boldsymbol{\phi'_s}^\top \mathrm{E}_c[\boldsymbol{\phi'_s} \,|\, \boldsymbol{s}]\big] = \Phi_D^\top \Xi_c \Phi_D - \Phi_D^\top P^\top \Xi_c P \Phi_D$, where the last term does not vanish in general.

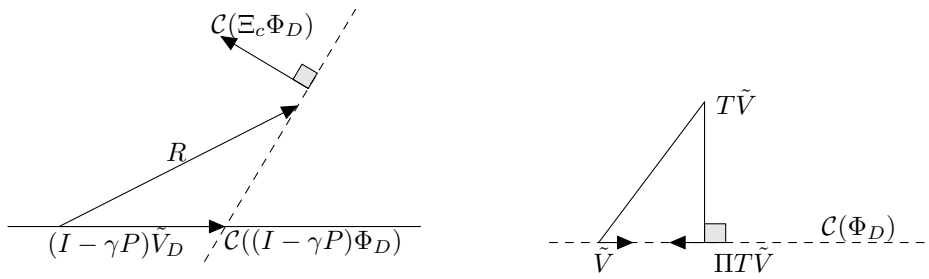

Figure 1: LSTD can be interpreted as an oblique projection (left) and as a fixpoint algorithm (right).

7 Two kinds of oblique projection

There is one more way to interpret the instrumental variable approach. Observe that the equation $\Pi_S r_S = \Pi_S D\Phi_S \hat{w}$, can be rewritten as $\Pi_S(D\Phi_S\hat{w} - r_S) = 0$. Thus we have that applying the projection amounts to solving $r_S = D\Phi_S\hat{w}$ under the constraint that the projection of the residual on the feature space is zero. Therefore LSTD yields the same solution as applying the oblique projection of the rewards on the difference between values of successive states (i.e. $D\Phi_S$), along the subspace orthogonal to the column space of Φ_S (which is the left null-space of Φ_S). See also figure 1.

Recall the formula for the coefficients of the oblique projection on the columns space of X orthogonal to the column space of Y, which is $(Y^\top X)^{-1}Y^\top$. It is easy to verify that putting $X = (I-\gamma P)\Phi_D$ and $Y = \Xi_c\Phi_D$ recovers the LSTD solution. Notice that in this case, the projected vector, $X(Y^\top X)^{-1}Y^\top$ corresponds to obtaining the 'smoothed rewards' corresponding to the approximate value function (i.e. $(I-\gamma P)\tilde{V}_D$, or what the rewards would have been if there had been no approximation of the value function). Now there is also a different way of defining the projection, namely we can project not the reward vector but the true value function [19]. In this case, setting $X = \Phi_D$ and $Y = (I-\gamma P)^\top \Xi_c\Phi_D$ again produces the LSTD solution w. Notice that in this case the projected vector corresponds to the approximate value function.

Notice that formally speaking, in both the interpretation as a projection of the reward vector and the value function, we also need another condition to call LSTD an oblique projection – in order for the formula $X(Y^\top X)^{-1}Y^\top$ to mean a projection on $\mathcal{C}(X)$ orthogonal to $\mathcal{C}(Y)$, we need the condition that the orthogonal complement of $\mathcal{C}(X)$ and $\mathcal{C}(Y)$ should be complementary subspaces. We will now claim that this is the case in either of the above ways of thinking about LSTD as a projection. To do this, we will prove the following statement, for any invertible matrices A,B, where we assume that A is invertible and $B\Phi_D$ is full column rank. We denote by k the number of columns in Φ_D (they are known to be linearly independent by assumption (A.1)).

$$\mathcal{C}(A\Phi_D)^\perp \oplus \mathcal{C}(B\Phi_D) = \mathbb{R}^n \quad \Leftrightarrow \quad \neg\exists z.\Phi_D^\top A^\top B\Phi_D z = 0$$

First, we note that the dimension of $\mathcal{C}(B\Phi_D)$ is k since $B\Phi_D$ is full column rank and the dimension of $\mathcal{C}(A\Phi_D)^\perp$ is exactly $n-l$ since A is invertible. The argument in the left-to-right direction is as follows: if $\exists z.\Phi_D^\top A^\top B\Phi_D z = 0$, then there would be a vector, $B\Phi_D z$, which is both in $\mathcal{C}(B\Phi_D)$ and $\mathcal{C}(A\Phi_D)^\perp$. Therefore these two subspaces cannot sum to the n-dimensional space if they share a common vector. This contradiction finishes the argument. The argument in the right-to-left direction is thus: there is no vector in both $\mathcal{C}(B\Phi_D)$ and $\mathcal{C}(A\Phi_D)^\perp$, then because of their dimensions they have to sum to the whole space $\mathbb{R}^n$.

We now see that the condition $\neg\exists z.\Phi_D^\top A^\top B\Phi_D z = 0$ is fulfilled in the case of LSTD because by assumption (A.1) the matrix $\Phi_D^\top A^\top B\Phi_D$, and hence also $\Phi_D^\top B^\top A\Phi_D$ has to be invertible. In this expression, we can substitute $A = I$ and $B = (I-\gamma P)^\top\Xi_c$ or alternatively $A = I-\gamma P$ and $B = \Xi_c$ to obtain either of the interpretations of LSTD as projection outlined above. We note that in either case, $B\Phi_D$ is full column rank by assumption (A.1) together with fact (F.1) and A is invertible since P is a Markov matrix.

8 Decompositions of the loss

We now present an interpretation of the minimization defined in equation 2, after [1]. We recall that the minimization in equation 2 can be rewritten in the following way $\Phi y^\star = \text{argmin}_y \|T\Phi y^\star - \Phi y\|_{\Xi_c} = \Pi(T(\Phi y^\star))$. Therefore $\Phi y^\star - \Pi(T(\Phi y^\star)) = 0$, or $\|\Phi y^\star - \Pi(T(\Phi y^\star))\|_{\Xi_c} = 0$. Therefore LSTD can be seen as the following.

$$y^\star = \operatorname*{argmin}_y \|\Phi y - \Pi(T(\Phi y))\|_{\Xi_c} \tag{11}$$

We note that this expression has no recursion and that the minimization is guaranteed to reach the optimum value of zero. We can now rewrite the norm as follows $\|\Phi y - \Pi(T(\Phi y))\|_{\Xi_c} = \|\Phi y - T(\Phi y)\|_{\Xi_c} -$

$\|\Pi(T(\Phi y)) - T(\Phi y)\|_{\Xi_c}$, where the equality follows from the Pythagorean theorem and the fact that $\Phi y - \Pi(T(\Phi y))$ and $\Pi(T(\Phi y)) - T(\Phi y)$ are orthogonal vectors, with respect to the Ξ_c-weighted dot product, which corresponds to Π. We thus obtain the following formula for the LSTD solution.

$$y^\star = \operatorname*{argmin}_y \|\underbrace{\Phi y - T(\Phi y)}_{\text{Bellman residual}}\|_{\Xi_c} - \|\Pi(T(\Phi y)) - T(\Phi y)\|_{\Xi_c} \tag{12}$$

We see that the LSTD algorithm minimizes a quantity which is the Bellman residual minus the reprojection error on the feature space. We discuss in section 9 the difference between simply minimizing the Bellman residual only and the LSTD algorithm.

Another way to interpret the LSTD loss is to see it as a nested optimization problem [11], which leads to the following two equivalent formulations. First, define the projection in the following way.

$$h^\star(y) = \operatorname*{argmin}_h \|\Phi h - T(\Phi y)\|_{\Xi_c} \tag{13}$$

Then we plug this for the definition of $\Pi(T(\Phi y))$ in equations 11 and 12 respectively, giving the following equivalent equations.

$$y^\star = \operatorname*{argmin}_y \|\Phi y - \Phi h^\star(y)\|_{\Xi_c} \quad \text{or} \quad y^\star = \operatorname*{argmin}_y \left(\|\Phi y - T(\Phi y)\|_{\Xi_c} - \|\Phi h^\star(y) - T(\Phi y)\|_{\Xi_c}\right) \tag{14}$$

9 The difference between LSTD and Bellman Residual Minimization

Instead of constructing the oblique projection as described in the previous sections, we can use a simpler algorithm, known as the Bellman Residual Minimization, which corresponds directly to projecting the rewards on the differences between successive states (see figure 2) – i.e. it is similar to LSTD except the projection is orthogonal, not oblique. BRM can be interpreted as the un-nested version of the optimization from the previous section.

$$h^\star = \operatorname*{argmin}_h \|\Phi h - T(\Phi h)\| \tag{15}$$

The reason LSTD was originally introduced as an improvement over BRM [6] is that for BRM, we do not have a justification in terms of a statistical model similar to the one we had in section 6 – the noise terms are correlated, so we cannot use a similar reasoning to claim consistency of BRM. But of course the fact that one line of deriving an algorithm doesn't work for BRM does not mean that the algorithm is wrong – there may be other justifications available. Interestingly, it can be shown that under our assumption (A.1) the two approaches are similar (the argument comes from chapter 4 of [2]). Indeed, we have from the previous section (compare equation 11) that LSTD is similar except for the presnce of the projection Π. It is sometimes useful to have formulas that make the difference between the two algorithms explicit in different formulations of each algorithm. The algebraic relationships between the two algorithms are summarized in the table below.

LSTD	BRM
$\min_w \|\Pi T\Phi w - \Phi w\|_{\Xi_c}$	$\min_w \|T\Phi w - \Phi w\|_{\Xi_c}$
$\min_w \|T\Phi w - \Phi w\|_{\Xi_c} - \|\Pi T\Phi w - T\Phi w\|_{\Xi_c}$	$\min_w \|T\Phi w - \Phi w\|_{\Xi_c}$
$w = \left(\Phi_D^\top \Xi_c L \Phi_D\right)^{-1} \Phi_D^\top \Xi_c r_D, \quad L = I - \gamma P$	$w = \left(\Phi_D^\top L^\top \Xi_c L \Phi_D\right)^{-1} \Phi_D^\top L^\top \Xi_c r_D$
$\min_{w'} \|V - \Phi_D w'\|_{(I-\gamma P)^\top \Pi^\top \Xi_c \Pi (I-\gamma P)}$	$\min_{w'} \|V - \Phi_D w'\|_{(I-\gamma P)^\top \Xi_c (I-\gamma P)}$
$\Phi w = \Pi T \Phi w$	$\Phi w = \underbrace{\Phi(\Phi^\top L^\top \Xi_c \Phi)^{-1} \Phi^\top L^\top \Xi_c}_{\text{oblique projection, see [19]}} T\Phi w$

There has been renewed interest in the analysis of the difference between the two algorithms. One argument [19] is that in an off-line setting (i.e. in the situation when the weighing coefficients are different from the stationary distribution of the MRP, a scenario we do not consider in this paper) a performance bound can be shown about BRM that is impossible to derive about LSTD [19], on the other hand LSTD remains widely used in practice.

There is yet one more feature that means that LSTD is preferable to BRM is some practical cases – while with LSTD, as we have shown above, we only need one sequence of samples of features of states and a sequence of samples of reward to obtain an estimate of the value function; but with BRM we need to have two samples of the features of states.

We will now show a way to obtain a sample-based estimate $\hat{w}_B$ of the BRM solution, based on section 3.1 of [16]. We want to minimize the expectation $\mathrm{E}_c\left[(\boldsymbol{\phi} w_B - \boldsymbol{\phi}' w_B - \boldsymbol{r})^2\right]$. We have the sampled features Φ_S^1 and the sampled rewards r_S. We also have a second set of sampled features Φ_S^2. The sampled features are produced using the following process: given the trajectory $s_1, s_2, \ldots$, the features in Φ_S^1

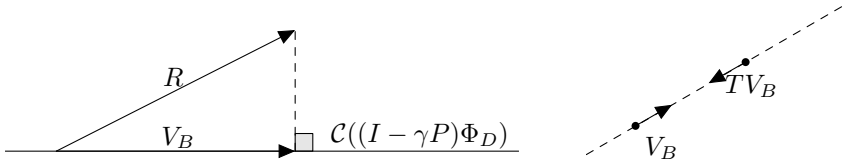

Figure 2: BRM as projection of rewards (left) and minimizing the Bellman residual (right). Cmp. fig. 1

are $\phi(s_1), \phi(s_2), \ldots$ while the features in Φ_S^2 correspond to 'alternative' states $s_2', s_3', \ldots$ sampled from $P(\cdot|s_1), P(\cdot|s_2), \ldots$. In other words, the features in Φ_S^2 describe where the MRP might also have gone to given a particular previous state. Of course, such sampling is only possible if we have a model of the transition dynamics of the MRP. Now, we can write a sample-based approximation to the expectation given above as $\hat{E} = \frac{1}{N-1}\sum_{i=1}^{N-1}(\Phi_S^1(i)w_B - \gamma\Phi_S^1(i+1)w_B - r_S(i))(\Phi_S^1(i)w_B - \gamma\Phi_S^2(i)w_B - r_S(i))$, where the notation $\Phi_S^1(i)$ means selecting row i of the matrix $\Phi_S^1(i)$ (i.e. the i-th feature in the trajectory). We can now introduce the notation $\Psi^1 = \Phi_S^1(1:N-1) - \gamma\Phi_S^1(2:N)$ and $\Psi^2 = \Phi_S^1(1:N-1) - \gamma\Phi_S^2(1:N)$, where the colon notation denotes ranges of rows. With this notation, we have that $w_B^\top {\Psi^1}^\top \Psi^2 w_B = w_B^\top {\Psi^2}^\top \Psi^1 w_B = \sum_{i=1}^{N-1}(\Phi_S^1(i)w_B - \gamma\Phi_S^1(i+1)w_B)(\Phi_S^1(i)w_B - \gamma\Phi_S^2(i)w_B)$. It can now be seen after a few rearrangements that $\hat{E} = \frac{1}{N-1}\left(w_B^\top {\Psi^1}^\top \Psi^2 w_B - r_S^\top(\Psi^1+\Psi^2)w_B + r_S^\top r_S\right) = \frac{1}{N-1}\left(\frac{1}{2}w_B^\top({\Psi^1}^\top\Psi^2 + {\Psi^2}^\top\Psi^1)w_B - r_S^\top(\Psi^1+\Psi^2)w_B + r_S^\top r_S\right)$. Taking the gradient with respect to w_B leaves the us with the system $({\Psi^1}^\top\Psi^2 + {\Psi^2}^\top\Psi^1)\hat{w}_B = (\Psi^1+\Psi^2)^\top r_S$, where we denoted by $\hat{w}_B$ the sample-based BRM solution.

10 Derivation using a linear dynamical system

This section is based on [18]. We will begin by constructing a MRP which lives in the space of features instead of our original state space. We limit ourselves to the class of linear dynamical systems. We need to define the matrix F and the vector q, so that a transition from $\boldsymbol{\phi}$ to $\boldsymbol{\phi}'$ (row vectors) is modelled by $\boldsymbol{\phi}F = \boldsymbol{\phi}'$, and the reward we expect at $\boldsymbol{\phi}$ is modelled by $\boldsymbol{\phi}q = \boldsymbol{r}$. Now we look for the values for F and q that model our system dynamics. We have that $\Phi_D F$ should be approximately equal to $P\Phi_D$ and $\Phi_D q$ to $\boldsymbol{r}$. We weigh states by Ξ_c, giving the following optimization problems.

$$F = \operatorname*{argmin}_F \|\Phi_D F - P\Phi_D\|_{\Xi_c} = \operatorname*{argmin}_F \operatorname{trace}\left((\Phi_D F - P\Phi_D)^\top \Xi_c (\Phi_D F - P\Phi_D)\right)$$
$$q = \operatorname*{argmin}_q \|\Phi_D q - r_D\|_{\Xi_c} = \operatorname*{argmin}_q (\Phi_D q - r_D)^\top \Xi_c (\Phi_D q - r_D) \tag{16}$$

These optimization problems correspond to ordinary least squares (generalized to matrices in case of F) and the solutions are obtained by projection: $\Phi_D F = \Pi P \Phi_D$ and $\Phi_D q = \Pi r_D$, where the projection matrix is defined as in section 5 ($\Pi = \Phi_D(\Phi_D^\top \Xi_c \Phi_D)^{-1}\Phi_D^\top \Xi_c$) and the matrix Φ_D cancels with the one in the projection, since it is full column rank. Now consider a feature vector ϕ. In the new MRP, we can compute the value function exactly (i.e. all the approximation has already taken place when we constructed the matrix F and vector q). The true value function associated with it is the expected discounted future reward: $\phi\sum_{i=0}^{\infty}(\gamma F)^i q$. Note that the vector of values corresponding to states of the original MRP is given by $\Phi_D\sum_{i=0}^{\infty}(\gamma F)^i q$. This is obviously a linear combination of columns of Φ_D (features) – this is why we introduced the projection in the first place. Thus we have $w = \sum_{i=0}^{\infty}(\gamma F)^i q = q + \sum_{i=1}^{\infty}(\gamma F)^i q = q + \gamma F w$, where the last equality is the well known telescoping sum argument. We thus have the equation $(I - \gamma F)w = q$, which is exactly the same as equation 4 once we plug in the computed values of F and q. Thus we have obtained the same estimator.

In the above, we assumed that the series $\sum_{i=0}^{\infty}(\gamma F)^i q$ converges. We show a stronger condition, namely that the series $\sum_{i=0}^{\infty}(\gamma F)^i$ converges. This follows from the following reasoning. We introduce the notation $\Pi^{\text{-}} = (\Phi_D^\top \Xi_c \Phi_D)^{-1}\Phi_D^\top \Xi_c$, so that $\Pi = \Phi_D \Pi^{\text{-}}$. Now we have that $F = \Pi^{\text{-}} P \Phi_D$, $q = \Pi^{\text{-}} r_D$ and $F^k = \Pi^{\text{-}} P(\Pi P)^{k-1}\Phi_D$ for $k \geq 1$. Therefore to have convergence, it suffices to establish that $\gamma\Pi P$ has spectral radius less than one. Consider first the case when we have $\Xi_c > 0$. Now, is is well known (see for example [4], proposition 6.3.1) that $\gamma\Pi P$ is a contraction in the norm weighted by Ξ_c; now, because we have $\Xi_c > 0$, this implies that the spectral radius condition holds. In the case where we have transient states, we use the following reasoning. We use the block matrix notation from section 19. Now, because P_{nt} in necessarily zero, it can be seen after a brief calculation that the matrix $\Pi\Phi_D$ has the following block matrix form.

$$\Pi P = \left[\begin{array}{c|c} \Pi_f P_f & 0 \\ \hline \cdots & 0 \end{array}\right] \quad \text{where} \quad \Pi_f = \Phi_D^f(\Phi_{fD}^\top {\Xi_c}^f \Phi_{fD})^{-1}\Phi_{fD}^\top {\Xi_c}^f$$

In the above, Φ_{fD} contains only the columns corresponding to recurrent states and the block matrix

denoted with $\cdots$ is not important for our reasoning. Now, because we have zeros in the second block column it can be seen that all eigenvalues of $\Pi\Phi_D$ are either zero or they are also the eigenvalues of $\Pi_f P_f$, where our spectral radius result holds by the contraction argument as before. Thus we have the result for the general case.

Note that in the above proof we used the fact that the distribution that Ξ_c has on the diagonal is a left eigenvector of P corresponding to eigenvalue one. If Ξ_c used for the projection were an *arbitrary* distribution, then the matrix F would in general have spectrum beyond the unit circle (see [8] for a concrete example).

We also stress the interpretation of the matrices F and q in terms of expectations, as given below.

$$F = (\Phi_D^\top \Xi_c \Phi_D)^{-1} \Phi_D^\top \Xi_c P \Phi_D = \mathrm{E}_c\left[\boldsymbol{\phi}_{\boldsymbol{s}}^\top \boldsymbol{\phi}_{\boldsymbol{s}}\right]^{-1} \mathrm{E}_c\left[\boldsymbol{\phi}_{\boldsymbol{s}}^\top \boldsymbol{\phi}_{\boldsymbol{s}}'\right]$$

$$q = (\Phi_D^\top \Xi_c \Phi_D)^{-1} \Phi_D^\top \Xi_c r_D = \mathrm{E}_c\left[\boldsymbol{\phi}_{\boldsymbol{s}}^\top \boldsymbol{\phi}_{\boldsymbol{s}}\right]^{-1} \mathrm{E}_c\left[\boldsymbol{\phi}_{\boldsymbol{s}}^\top \boldsymbol{r}\right]$$

We wish to note the applicability of the described definitions of F and q to the setting where we have a single transition matrix P, but instead of just one reward we have many tasks, each of which with a different reward [15]. We note that in this setting, while we still have to learn q for each task separately, it is worthwhile to learn F using training data from *all* tasks.

We also want to stress the fact that we can construct sample-based variants of the matrices F and q (call them $\hat{F}$ and $\hat{q}$ respectively) and still obtain the same algorithm. Let us adopt the following definitions.

$$\hat{F} = \operatorname*{argmin}_{\hat{F}} \|\Phi_S \hat{F} - N\Phi_S\| = (\Phi_S^\top \Phi_S)^{-1}(\Phi_S^\top N \Phi_S)$$

$$\hat{q} = \operatorname*{argmin}_{\hat{q}} \|\Phi_S \hat{q} - r_S\| = (\Phi_S^\top \Phi_S)^{-1}(\Phi_S^\top r_S)$$

In the above, we denote by N the matrix which has ones above the main diagonal and zeros elsewhere. We note that the required inverse exists by assumption (A.1). It can be easily verified that the estimate $\hat{w} = (I - \gamma\hat{F})^{-1}\hat{q}$ is consistent with what we gave in the section about instrumental variables.

11 Derivation from the iterative TD algorithm

We have seen in section 6 that the equality $\mathrm{E}_c\left[\boldsymbol{\phi}^\top e_{\boldsymbol{s}}\right] = -\Phi_D^\top \Xi_c r_D + \Phi_D^\top \Xi_c (I - \gamma P)\Phi_D w = 0$ is crucial for the development of the algorithm and indeed equivalent to the obtained estimator for w (equation 5). We will now show another way of obtaining this equality – actually, it may be taken do be the *definition* of the algorithm, and used as a justification for the formula 5 that stands on its own. We now give the interpretation of this equation is in terms of the iterative TD algorithm [23]. We note that the equality $0 = \mathrm{E}_c\left[\boldsymbol{\phi}^\top e_{\boldsymbol{s}}\right]$ corresponds to saying that the LSTD solution corresponds to the fixpoint of iterative TD, i.e. the point where the expected update is zero.

Consider now the definition of the iterative TD algorithm [23]. We assume for the moment that we have an oracle for the value function and are interested in iteratively solving the optimization problem $\min_w (V_o(\boldsymbol{s}) - \tilde{V}(\boldsymbol{s}))^2$ using the approcimation architecture $\tilde{V}(\boldsymbol{s}) = \boldsymbol{\phi}_{\boldsymbol{s}} w$. The iterative update is given by $\nabla_w (V_o(\boldsymbol{s}) - \tilde{V}(\boldsymbol{s}))^2 = 2\nabla_w \tilde{V}(\boldsymbol{s})\left(V_o(\boldsymbol{s}) - \tilde{V}(\boldsymbol{s})\right)$. We now have the following formula for the iteration.

$$\Delta w \propto \underbrace{\nabla_w \tilde{V}(\boldsymbol{s})}_{\boldsymbol{\phi}(\boldsymbol{s}_t)^\top} \underbrace{\left(\underbrace{\left(\boldsymbol{r}_{t+1} + \gamma \tilde{V}(\boldsymbol{s}_{t+1})\right)}_{\text{oracle for value}} - \tilde{V}(\boldsymbol{s}_t)\right)}_{\text{TD error } e_{\boldsymbol{s}}}$$

Now we have that the update Δw at time t, is $\boldsymbol{\phi}(\boldsymbol{s}_t)^\top e_{\boldsymbol{s}_t}$. Setting the expectation of this update to zero gives the desired formula. We also note that the relation between the TD iteration and the LSTD algorithm resembles the chicken-and-egg problem – one can either, as we did above, consider the iteration a priori knowledge and use that to justify the LSTD fixpoint, or one can start with the fixpoint and treat the iteration as a way of reaching it, motivated by stochastic optimization. LSTD can also be extended to compute the fixpoints of TD(λ) or, more generally other similar algorithms with different traces. For details, see [7] in slightly different notation.

12 Interpretation in terms of minimizing a quadratic form

This section is based on [22]. It interprets LSTD as the minimization of a quadratic form in the error between the true value function $V(\cdot)$ and the approximated value function $\Phi_D w$. We begin by reformulating the formula for the estimator obtained above.

$$w = \left(\Phi_D^\top \Xi_c (I - \gamma P)\Phi_D\right)^{-1} \Phi_D^\top \Xi_c r_D =$$
$$= \left(\Phi_D^\top (I - \gamma P)^\top \Xi_c \Phi_D (\Phi_D^\top \Xi_c \Phi_D)^{-1} \Phi_D^\top \Xi_c (I - \gamma P)\Phi_D\right)^{-1} \Phi_D^\top (I - \gamma P)^\top \Xi_c \Phi_D (\Phi_D^\top \Xi_c \Phi_D)^{-1} \Phi_D^\top \Xi_c (I - \gamma P) V$$

This equality holds because $r_D = (I - \gamma P)V$ and because the matrices $\Phi_D^\top (I - \gamma P)^\top \Xi_c \Phi_D$ and $\Phi_D^\top \Xi_c \Phi_D$ are invertible by assumption (A.1). Now, we introduce the matrix K, as below.

$$K = (I - \gamma P)^\top \Xi_c \Phi_D (\Phi_D^\top \Xi_c \Phi_D)^{-1} \Phi_D^\top \Xi_c (I - \gamma P) = (I - \gamma P)^\top \Pi^\top \Xi_c \Pi (I - \gamma P)$$

We note that $\Xi_c \Phi_D (\Phi_D^\top \Xi_c \Phi_D)^{-1} \Phi_D^\top \Xi_c = \Xi_c \Pi = \Pi^\top \Xi_c = \Pi^\top \Xi_c \Pi$, where the last equality follows by substituting the definition of Π and canceling the inverted term. Therefore we have $w = \left(\Phi_D^\top K \Phi_D\right)^{-1} \Phi_D^\top K V$. But this is the solution to the well-known optimization problem: $w = \text{argmin}_{w'} \|V - \Phi_D w'\|_K = \text{argmin}_{w'} (V - \Phi_D w')^\top K (V - \Phi_D w')$. Thus we gain an insight about approximation $\tilde{V}(\cdot)$ of equation 6 – instead of minimizing the norm $\|\cdot\|_{\Xi_c}$, which would yield us $\bar{V}_D$, we minimize the different norm $\|\cdot\|_K$, thus gaining the ability of efficiently estimating the solution from samples. Note that we can also repeat the above reasoning, without the multiplication by $(\Phi_D^\top \Xi_c \Phi_D)^{-1}$, to obtain the matrix $K' = (I - \gamma P)^\top \Xi_c \Phi_D \Phi_D^\top \Xi_c (I - \gamma P)$ which also defines a valid minimization – this is the way the equivalence was originally introduced in [20].

13 LSTD is a subspace algorithm

In section 7, we have shown that the algorithm can be thought of as an oblique projection along the subspace orthogonal to the feature space. Here, we make explicit the property that LSTD only depends on the features through the subspace they span i.e. any full-rank transformation (i.e. basis change) C of features does not influence the value function. To see this, consider the sample estimate we derived in earlier sections, where we use the transformed features $\Phi_S C$ instead of Φ_S.

$$\begin{aligned}\hat{V}_C &= \Phi_S C \hat{w}_C = \Phi_S C (C^\top \Phi_S^\top D \Phi_S C)^{-1} C^\top \Phi_S^\top r_S = \\ &= \Phi_S C C^{-1} (\Phi_S^\top D \Phi_S)^{-1} {C^\top}^{-1} C^\top \Phi_S^\top r_S = \Phi_S (\Phi_S^\top D \Phi_S)^{-1} \Phi_S^\top r_S = \hat{V}\end{aligned}$$

As a corollary, we state that LSTD is independent of any scaling of features.

14 The case with the singular matrix

We are now concerned with the case where the assumption (A.1) is not met, i.e. $\mathrm{E}_c\left[\boldsymbol{\phi}^\top (\boldsymbol{\phi} - \gamma \boldsymbol{\phi}')\right]$ is not invertible. It then follows by fact (F.1) that the matrix $\mathrm{E}_c\left[\boldsymbol{\phi}^\top \boldsymbol{\phi}\right]$ is also not invertible. We will now argue for that in such a case, taking the pseudo-inverse instead of the inverse of the matrix $\mathrm{E}_c\left[\boldsymbol{\phi}^\top (\boldsymbol{\phi} - \gamma \boldsymbol{\phi}')\right]$ in the LSTD formula produces a good estimate of the value function. To do this, we will discuss two cases in which the matrix $\mathrm{E}_c\left[\boldsymbol{\phi}^\top \boldsymbol{\phi}\right]$ may be singular. In writing this section we had to assume a conjecture, outlined in section 21, which we strongly suspect is true but have no proof for it.

First, it may be the case that Φ_D has dependent columns. We also assume that $\Xi_c > 0$ (we will deal with the case where this is not the case in the next section). We introduce the matrix C, which selects a basis from the columns of Φ_D, so that $\Phi_D C$ has independent columns and C is a skinny matrix with columns from the identity matrix. We note that the matrix $C^\top \Phi_D^\top \Xi_c \Phi_D C$ is invertible. We now wish to prove that $V_C = \Phi_D C (C^\top \Phi_D^\top \Xi_c (I - \gamma P) \Phi_D C)^{-1} C^\top \Phi_D^\top \Xi_c r_D = \Phi_D (\Phi_D^\top \Xi_c (I - \gamma P) \Phi_D)^+ \Phi_D^\top \Xi_c r_D = V_+$. Now if we assume that our conjecture about oblique projection holds (see section 21) we can see that this follows automatically since $\mathcal{C}(\Phi_D) = \mathcal{C}(\Phi_D C)$, $\mathcal{C}((I - \gamma P)^\top \Xi_c \Phi_D) = \mathcal{C}((I - \gamma P)^\top \Xi_c \Phi_D C)$.

Having examined this situation, we now assume that the columns of Φ_D are independent. Algebraically speaking it could happen that for some vector $v \neq 0$, we have $\Phi_D v \neq 0$ but $\|\Phi_D v\|_{\Xi_c} = 0$, since the notation $\|\cdot\|_{\Xi_c}$ only denotes a *semi*-norm, i.e. the vector $\Phi_D v$ is non-zero only in states to which Ξ_c assigns weight zero. We will now argue that this is impossible given how we construct our matrix. Consider the matrix E whose rows are rows of the identity matrix which eliminates the transient states, so that we have $\Phi_D^R = E \Phi_D$, where Φ_D^R is the same as Φ_D except it only contains rows corresponding to the recurrent states. We similarly have $\Xi_c^R = E \Xi_c E^\top$. Now, we can see from the interpretation of the matrix $\Phi_D^\top \Xi_c \Phi_D$ as the expectation $\mathrm{E}_c\left[\boldsymbol{\phi}^\top \boldsymbol{\phi}\right]$ that simply removing the states with zero weights (the transient states) does not change the value of the expression, so we have $\Phi_D^\top \Xi_c \Phi_D = {\Phi_D^R}^\top \Xi_c^R \Phi_D^R$. But the second matrix is invertible.

15 Regularization

To overcome the problem of over-fitting, the standard procedure is to add a regularization term to the proposed algorithm. There are many ways of doing that.

One way, proposed by [14] is to consider the optimization problem of the fixpoint equation 2. We can extend it as follows: $\Phi_D w = \text{argmin}_{w'} \left(\|r_D + \gamma P \Phi_D w - \Phi_D w'\|_{\Xi_c} + \beta \|w'\|\right)$. Here, $\beta \geq 0$ is an external parameter of the algorithm and both norm expressions are the usual L_2 norm, the first one weighted by Ξ_c. This way of regularizing produces the well-known analytic solution $w_R = \left(\Phi_D^\top \Xi_c (I - \gamma P) \Phi_D + \beta I\right)^{-1} \Phi_D^\top \Xi_c r_D$. In the paper [14], a version is also given where the second norm is L_1. In this case, because equation 2 is a fix-point equation, it is not possible to simply plug the problem into the standard LASSO algorithm, and a new algorithm is necessary (see [14] for details).

Before we continue, denote the standard L_2-regularized solution of a system of equations $Ax = b$ as $\text{solve}_{L_2}(A, b, \beta) = \text{argmin}_x \|Ax - b\|_{\Xi_c} + \beta\|x\|_2 = (A^\top \Xi_c A + \beta I)^{-1} A^\top \Xi_c b$. Denote the version with L_1 regularization as $\text{solve}_{L_1}(A, b, \beta) = \text{argmin}_x \|Ax - b\|_{\Xi_c} + \beta\|x\|_1$ (this has no explicit analytic form as has to be computed using an algorithm, typically LASSO). A second way of regularization, introduced in [11] is to add regularization to equation 14, giving the following optimization problem.

$$y^\star = \underset{y}{\text{argmin}} \|\Phi y - \Phi h^\star(y)\|_{\Xi_c} + \|y\|_{1 \text{ or } 2}$$

In the above, the final norm may be either of L_2 or L_1. A quick calculation shows that this is the same as regularizing the system of equations 4. This idea therefore corresponds to the solutions $\text{solve}(\Phi_D(I - \gamma(\Phi_D^\top \Xi_c \Phi_D)^{-1}\Phi_D^\top \Xi_c P \Phi_D)), \Phi_D(\Phi_D^\top \Xi_c \Phi_D)^{-1}\Phi_D^\top \Xi_c r_D, \beta)$ for each of the discussed norms.

Another way is adding regularization directly to the equation where we have already solved for w, that is, $w = A^{-1}b$, where $A = \left(\Phi_D^\top \Xi_c (I - \gamma P)\Phi_D\right)$ and $b = \Phi_D^\top \Xi_c r_D$. If we regularize with L_2, this corresponds to the solutions $\text{solve}_{L_2}(A, b, \beta)$. This (together with other versions, that do not map to LSTD), has been done in [2], where the author also derives finite-sample error bounds.

It is also possible to combine some of the above ways together, after the manner of [13], and to use other sparsifiers in place of L_1. In [12], for instance, the Dantzig selector is employed, which leads to a considerable simplification of the optimization problem (the optimization reduces to a linear program).

A yet different approach [17] to regularization is to keep the algorithm itself unchanged and instead do feature selection beforehand. Even if the feature selection algorithm is very simple (greedy based on correlation with residual), simulations [17] suggest that doing feature selection leads to performance essentially the same as the approaches described above. Because greedy feature selection is so simple, this suggests that regularization of LSTD is not yet really a fully solved problem.

We now describe a different way of regularization. The idea is to consider the linear-system interpretation of LSTD and to regularize the linear system. A simple way of doing this, proposed in [5], is to add an additional rank constraint (over the matrix F) to the minimization of equation 16, to obtain the following.

$$\underset{F}{\text{argmin}} \|\Phi_D F - P\Phi_D\|_{\Xi_c} \quad \text{subject to} \quad \text{rank}(F) \leq k \tag{17}$$

This regularized F does not have a closed form any longer, but can still be obtained efficiently using SVD, in the following way. In the notation of section 20.2, we have that $X = \Phi_D$ and $Y = P\Phi_D$. Denote as U, Λ, V the singular value decomposition of the matrix ${\Xi_c}^{1/2}\Phi_D({\Xi_c}^{1/2}\Phi_D)^+ {\Xi_c}^{1/2} P\Phi_D = {\Xi_c}^{1/2}\Pi P \Xi_c \Phi_D$. It can be shown (see [9] and 20.1, that the best rank-k approximation $\tilde{F}$ is given by the formula $FVS_kV^\top$, where F is the full rank solution, the diagonal matrix S_k has ones in the first k diagonal entries and zeros elsewhere. Observe that this is equivalent to leaving just the first k columns of the matrix V (call the resulting sub-matrix V_k) and using the formula $FV_kV_k^\top$. This formula has an interesting interpretation – it can be seen as a product of two rectangular matrices $(FV_k)(V_k^\top)$, the first one skinny and the other fat. The process of approximating the next feature using this matrix can be thought to consist of two stages: compressing the feature vector into a shorter one (of length k) and then decompressing it again to full size. Notice that the compression / decompression operators are not defined uniquely, even if all the singular values of the matrix ${\Xi_c}^{1/2}\Pi P \Xi_c \Phi_D$ are distinct – we can insert any basis change matrix B and still obtain a valid decomposition $(FV_kB)(B^{-1}V_k^\top)$. In short, this regularization procedure doesn't really compute latent (or compressed) *features*, but rather it computes the latent *subspace*.

Here, we have given the design (or limiting) variant of the regularized algorithm, i.e. the one that uses the formal specification of the MRP. It is of course possible o give a sample-based version. This can be done as follows. We start with the fllowing problem.

$$\hat{F} = \underset{\hat{F}}{\text{argmin}} \|\Phi_S \hat{F} - N\Phi_S\| \quad \text{subject to} \quad \text{rank}(\hat{F}) \leq k$$

We can then apply the mechanics described in section 20.1. We note that the rank constraint only affects the construction of $\hat{F}$, not $\hat{q}$.

The idea described in the previous paragraphs is not to be confused with simply performing the PCA in the feature space before invoking the algorithm, which amounts to first computing the SVD of the matrix ${\Xi_c}^{1/2}\Phi_D = U\Lambda V^\top$, to obtain the matrix V_k which contains the first k columns of V and then constructing the compressed features as $\Phi_D V_k V_k^\top$. This process, however, does not take into account the dynamics of the MDP – we see that none of the quantities involved depend in any way on the matrix P.

A property of all the above regularizers is that we lose the invariance of the algorithm w.r.t. the choice of basis for the feature space, which can be seen as a natural characteristic of LSTD[3]. It is not clear whether the property would be worth preserving in a regularized version – sparsity by its very nature is not invariant to transformations of features, even linear ones and there is a general tendency that a more specialized algorithm will have less generic properties.

[3]Indeed section 4 of [13] deals with how to perform standardization of features before plugging them into optimization.

16 An attempt to derive the asymptotic covariance of the computed estimate

This section is based on the standard method for deriving covariances in the instrumental variable setting, as described in [24]. Unfortunately, this method does not lead to a usable covariance estimate in our case. It is nonetheless instructive to look at how the derivation would work and where it breaks. We assume $\Xi_c > 0$, which is also the condition for consistency. In the preceding sections, we have obtained the estimator for w as $\mathrm{E}_c\left[\boldsymbol{\phi}^\top(\boldsymbol{\phi}-\gamma\boldsymbol{\phi}'_{\boldsymbol{s}})\right]^{-1}\mathrm{E}_c\left[\boldsymbol{\phi}^\top \boldsymbol{r}_{\boldsymbol{s}}\right]$. The sample-based equivalent $\hat{w}$ is $(\Phi_S^\top D\Phi_S)^{-1}\Phi_S^\top r_S = (\Phi_S^\top D\Phi_S)^{-1}\Phi_S^\top\left(D\Phi_S w + \boldsymbol{\zeta}_S - e_S\right) = w + (\Phi_S^\top D\Phi_S)^{-1}\Phi_S^\top(\boldsymbol{\zeta}_S - e_S)$. Now, we are interested in estimating the covariance structure of $\hat{w}$. We need to make the additional assumption that $\mathrm{E}_c\left[\boldsymbol{r}_{\boldsymbol{s}}^2\right] < \infty$, i.e. variance of the reward process exists (A.3). Note that this implies that $\mathrm{E}_c\left[\boldsymbol{\zeta}_{\boldsymbol{s}}^2\right] < \infty$. Now consider the distribution of the term $N^{-1/2}\Phi_S^\top(\boldsymbol{\zeta}_S - e_S)$. We cannot apply the central limit theorem (CLT) directly, because by the dynamics of the MRP, the elements in the sum $\Phi_S^\top\boldsymbol{\zeta}_S$ are not independent samples. Therefore we now rearrange the terms to obtain the following.

$$N^{-1/2}\Phi_S^\top(\boldsymbol{\zeta}_S - e_S) = N^{-1/2}\sum_{i=1}^N \boldsymbol{\phi}_i(\boldsymbol{\zeta}_{\boldsymbol{s}_i} - e_{\boldsymbol{s}_i}) = N^{-1/2}\sum_{s=1}^S \boldsymbol{\phi}_s\left(\sum_{i:\boldsymbol{s}_i=s}\boldsymbol{\zeta}_{\boldsymbol{s}_i}\right) - N^{-1/2}\sum_{i=1}^N \boldsymbol{\phi}_i e_{\boldsymbol{s}_i} =$$
$$= N^{-1/2}\sum_{s=1}^S c_s^{1/2}\boldsymbol{\phi}_s \underbrace{\left(\sum_{i:\boldsymbol{s}_i=s}\boldsymbol{\zeta}_{\boldsymbol{s}_i}\right)c_s^{-1/2}}_{\text{goes to } N(0,\,\mathrm{E}_c[\boldsymbol{\zeta}_{\boldsymbol{s}}^2])} - \underbrace{N^{-1/2}\sum_{i=1}^N \boldsymbol{\phi}_i e_{\boldsymbol{s}_i}}_{\text{goes to } N(0,\,\mathrm{E}_c\left[\boldsymbol{\phi}^\top\boldsymbol{\phi}e_{\boldsymbol{s}}^2\right])}$$

In the above, we applied the CLT to the sum of $\boldsymbol{\zeta}_{\boldsymbol{s}_i}$ in each state separately, since, in a given state, they are independent. The terms c_s stand for the expected frequencies of visiting each state, $c_s = N\xi_s$. So we have that the distribution of $N^{-1/2}\Phi_S^\top\boldsymbol{\zeta}_S$ converges to a Gaussian with mean zero and covariance $\sum_{s=1}^S \boldsymbol{\phi}_s^\top\boldsymbol{\phi}_s\,\mathrm{E}_c\left[\boldsymbol{\zeta}_{\boldsymbol{s}}^2\right]c_s N^{-1} \to \mathrm{E}_c\left[\boldsymbol{\zeta}_{\boldsymbol{s}}^2\boldsymbol{\phi}^\top\boldsymbol{\phi}\right]$ (note that this is the same formula we would have got if we had wrongly assumed samples in first sum were i.i.d. and applied the CLT). In the second term, we can apply the CLT because the sum is equivalent to drawing from a multinomial distribution corresponding to the states of the MRP, where the probabilities are given by the stationary distribution and the mean $\mathrm{E}_c[\boldsymbol{\phi}e_{\boldsymbol{s}}]$ is zero because of the property we discussed at the beginning of section 6.

Unfortunately, the argument breaks because we cannot claim that $N^{-1/2}\Phi_S^\top(\boldsymbol{\zeta}_S - e_S)$ converges to a Gaussian with mean 0 and covariance B given by $B \to \mathrm{E}_c\left[(\boldsymbol{\zeta}_{\boldsymbol{s}}^2 + e_{\boldsymbol{s}}^2)\boldsymbol{\phi}^\top\boldsymbol{\phi}\right]$ – we cannot assume anything about the dependence of the Gaussians that come from our two applications of the CLT. On the other hand, if we had been able to obtain the desired covariance B, we could assume (heuristically, but as is customary in asymptotic covariance derivations) that $N^{-1/2}(w - \hat{w})$ is normally distributed with covariance $\mathrm{E}_c\left[\boldsymbol{\phi}^\top(\boldsymbol{\phi}-\gamma\boldsymbol{\phi}'_{\boldsymbol{s}})\right]^{-1} B\,\mathrm{E}_c\left[(\boldsymbol{\phi}-\gamma\boldsymbol{\phi}'_{\boldsymbol{s}})^\top\boldsymbol{\phi}\right]^{-1}$. This means we could (in the asymptotic limit) treat $\hat{w}$ as normally distributed with the following covariance.

$$N\,\mathrm{E}_c\left[\boldsymbol{\phi}^\top(\boldsymbol{\phi}-\gamma\boldsymbol{\phi}'_{\boldsymbol{s}})\right]^{-1} B\,\mathrm{E}_c\left[(\boldsymbol{\phi}-\gamma\boldsymbol{\phi}'_{\boldsymbol{s}})^\top\boldsymbol{\phi}\right]^{-1}$$

The expectation term is easy to estimate from samples (and we need to compute it anyway to get the estimate of the value function). The problem is in the estimation of B. If we had access to an estimate $\hat{B}$, this would give rise to the following estimate for the covariance of $\hat{w}$.

$$\hat{\Sigma} = (\Phi_S^\top D\Phi_S)^{-1}\hat{B}(\Phi_S^\top D^\top\Phi_S)^{-1}$$

Given the design matrix Φ_D, we could then compute the covariance of the value function as $\Phi_D\hat{\Sigma}\Phi_D^\top$. In practice, we are unlikely to want to compute this using the whole design matrix. Instead, we will pick a set of interesting states and use a matrix whose rows correspond to them. Unfortunately, we don't know how to compute an expression for B, let alone an expression for $\hat{B}$, estimable from samples.

17 Bayesian interpretation of covariance

We now follow the analysis of [10] and define one possible probabilistic model for LSTD. We start with equation $(\Phi_S^\top D\Phi_S)\hat{w} = \Phi_S^\top r_S$. Consider an arbitrary symmetric matrix G. We can multiply both sides with $(\Phi_S^\top D\Phi_S)^\top G$ to obtain the following.

$$(\Phi_S^\top D\Phi_S)^\top G(\Phi_S^\top D\Phi_S)\hat{w} = (\Phi_S^\top D\Phi_S)^\top G\Phi_S^\top r_S$$

The $\hat{w}$ which solves this equation is the one which minimizes the following quadratic form.

$$\min_{\hat{w}} \|r_S - D\Phi_S\hat{w}\|_{\Phi_S G\Phi_S^\top}$$

This is the same as the Gaussian maximum likelihood problem where $P(r_S|\hat{w})$ is proportional to $\exp\left\{-\frac{1}{2}\cdot\right\}$ of the above, i.e. with the precision matrix $\Phi_S G\Phi_S^\top$. Now we can either solve this directly or we may optionally introduce a prior over $\hat{w}$ (assume a Gaussian prior with mean zero and precision L), to obtain the following.

$$P(r_S|\hat{w}) \propto \exp\left\{-\frac{1}{2}\|r_S - D\Phi_S\hat{w}\|_{\Phi_S G\Phi_S^\top} + \|\hat{w}\|_L\right\}$$

Now, if we take as our matrix G the hypothetical inverse covariance from the previous section, we obtain the interpretation [21] that the variance of our values is the mean projected Bellman error. This is appealing because it models our intuition that the projected Bellman error reflects our uncertainty about state values. But it should be noted that this correspondence is essentially *defined* and not *derived* in the preceding argument. It is not a deeper fact about the LSTD algorithm – indeed we could plug any other covariance matrix as G and obtain a similar result.

18 The episodic version of LSTD

In the other sections of this paper, we have considered the case where the MRP never terminates and convergence is defined by taking the limit with respect to the length of a trajectory. We are now interested in extending our observations to the case where there is a termination state. The limit will now be the with respect to the number of episodes being accumulated. First, let us note that the formula $w = \mathrm{E}_c\left[\boldsymbol{\phi}^\top(\boldsymbol{\phi} - \gamma\boldsymbol{\phi}'_{\boldsymbol{s}})\right]^{-1}\mathrm{E}_c\left[\boldsymbol{\phi}^\top \boldsymbol{r}_{\boldsymbol{s}}\right]$ is still valid in this case. We simply have to give new meaning to the expectation terms.

We will now start by giving a design-based variant for the algorithm. All transitions in a terminating MRP can be described using a rectangular matrix P_t, where the last column is meant to denote termination. We assume in the following that the starting state of the MRP is the first state. We also assume that the matrix P_t is such that the MRP will always eventually terminate. We first need to construct a state distribution Ξ. To do this, we append the row $[1, 0 \dots 0]$ to the matrix P_t, producing the square matrix P_a, which assumes that the MRP restarts after reaching the termination state. Now, the diagonal entries of the matrix Ξ are the entries of the left eigenvector of P_a which corresponds to eigenvalue one. Now we also construct another square matrix, P, which we obtain by appending the row $[0 \dots 0, 1]$ to the matrix P_t. This matrix assumes that the agent stays in the termination state forever. The intuition behind this is the following: the matrix P describes the true dynamics of the MRP, but in order to have a meaningful state distribution we need to take into account the fact that we have multiple episodes – hence the definition of the matrix P_a, which models restart. Having defined the above matrices, we may use the standard formula in the following way: $w = \left(\Phi_D^\top \Xi(I - \gamma P)\Phi_D\right)^{-1}\Phi_D^\top \Xi r_D$. Here, we assume that the last feature vector (i.e. the one corresponding to the state modelling termination) is zero. By definition, the final element of r_D is also zero.

It can be seen that the sample-based variant is the same as in the case of one long trajectory, except for the additional summation over the episodes. We note we use here the fact that the termination state has the feature of zero (so that we can still use the matrix D – there is no subtraction in the last row, but it doesn't matter since the last state is the terminal state). The formula looks as follows, where the sum goes over episodes.

$$\hat{w} = (\textstyle\sum_e \Phi_{S_e}^\top D_{S_e} \Phi_{S_e})^{-1}(\sum_e \Phi_{S_e}^\top r_{S_e})$$

19 A fact about assumption (A.1)

We prove that (A.1) implies (F.1). We rewrite them in matrix form: $\det(\Phi_D^\top \Xi_c \Phi_D) \neq 0$ and $\det(\Phi_D^\top \Xi_c (I - \gamma P)\Phi_D) \neq 0$. We will now develop the second expression. By the well-known eigenvalue argument, $I - \gamma P$ is invertible. Assume for the moment $\Xi_c > 0$ (we will deal with the case when this is not true later). Consider some non-zero vector x. We have that $\Phi_D^\top \Xi_c (I - \gamma P)\Phi_D x = 0$ if and only if the vector $y = \Phi_D x$, which in the column space of Φ_D satisfies the condition that $\Xi_c(I - \gamma P)y$ is orthogonal to the column space of Φ_D. This implies that $y^\top \Xi_c (I - \gamma P)y = 0$. This holds if and only if $y^\top(\frac{1}{2}(\Xi_c(I - \gamma P)) + \frac{1}{2}(\Xi_c(I - \gamma P))^\top)y = 0$. Now because the matrix defining this quadratic form is symmetric, and thus diagonalizable and with real eigenvalues, we have that this can only be zero if some of the eigenvalues are nonpositive. We will show that this cannot be the case. Rewrite the matrix $\frac{1}{2}(\Xi_c(I - \gamma P)) + \frac{1}{2}(\Xi_c(I - \gamma P))^\top$ as $\Xi_c(I - \gamma\frac{1}{2}(P + {\Xi_c}^{-1}P^\top \Xi_c))$. Now because by definition $\Xi_c = \mathrm{diag}(\xi)$ where $\xi^\top P = \xi^\top$, we have that ${\Xi_c}^{-1}P^\top \Xi_c[1, \dots, 1]^\top = [1, \dots, 1]^\top$; moreover, ${\Xi_c}^{-1}P^\top \Xi_c$ has positive entries. So it is a Markov matrix. Thus $\frac{1}{2}(P + {\Xi_c}^{-1}P^\top \Xi_c)$ also is a Markov matrix. Thus, $(I - \gamma\frac{1}{2}(P + {\Xi_c}^{-1}P^\top \Xi_c))$ has eigenvalues in the positive real half-plane. But this matrix is obviously similar to $\Xi_c(I - \gamma\frac{1}{2}(P + {\Xi_c}^{-1}P^\top \Xi_c))$ by the transformation ${\Xi_c}^{-1/2}$, so it has the same eigenvalues. This finishes the proof for $\Xi_c > 0$. Now consider the case when we do not have this, i.e. some of the diagonal entries of Ξ_c are zero. Intuitively, the fact we prove is now obvious since transient states do not influence the values of the expectations, except when the features of non-transient states are linearly dependent, which would violate (A.1). More formally, we can, without loss of generality assume that the states for which the probability given by the stationary distribution is zero have highest indexes (i.e. they occur at the back of matrices Ξ_c, P and Φ_D). We introduce the following notations for block minors of matrices Ξ_c, P and the vector y corresponding to the non-transient and transient states.

$$\Xi_c = \left[\begin{array}{c|c} {\Xi_c}^f & 0 \\ \hline 0 & 0 \end{array}\right] \quad P = \left[\begin{array}{c|c} P_f & P_{nt} \\ \hline P_{tn} & P_{tt} \end{array}\right] \quad y = \left[\begin{array}{c} y_f \\ \hline y_t \end{array}\right]$$

Note that in the above, P_{nt} is has to be the zero matrix – it corresponds to transitions from non-transient states to transient states. Therefore we have that $\Xi_c(I-\gamma P)y = 0$ implies ${\Xi_c}^f(I-\gamma P_f)y_f$ and thus, by the reasoning for the case without transient states, y_f has to be the zero vector, which violates assumption (A.1). Thus $\Xi_c(I-\gamma P)y$ is nonzero, for all nonzero vectors y in the column space of Φ_D. Therefore for $\Phi_D^\top \Xi_c(I-\gamma P)\Phi_D x$ to be zero, we need the vector $\Xi_c(I-\gamma P)\Phi_D x$ to be orthogonal to the column space of Φ_D. In particular, this implies $y^\top \Xi_c(I-\gamma P)y = 0$. Again, this simplifies to $y_f^\top {\Xi_c}^f(I-\gamma P_f)y_f = 0$. Again, by the reasoning for the case without transient states, this implies that y_f is the zero vector, which violates assumption (A.1). Thus we have the desired result.

20 Summary of results about reduced-rank regression

In the following sections, we summarize the results from literature concerning the algorithm for obtaining a solutions for reduced rank regression problems.

20.1 Reduced Rank Regression as as two-stage procedure

We first give the two-stage argument as outlined in [9]. Consider the optimization problem $\text{argmin}_F \|XF-Y\|$. The full-rank solution is $F_f = X^+Y$, which corresponds to the approximation $Y_f = XF_f = XX^+Y$. Consider now the SVD of $Y_f = U\Lambda V^\top$, where we assume that the singular values are decreasing. A reduced-rank approximation to Y_f of rank s can be obtained by stetting the smallest singular values to zero, leaving only s singular values, which yields the matrix $Y_s = U\Lambda I_s V^\top$, where we denote by I_s the matrix with ones in the first s diagonal entries and zeros elsewhere. Of all the matrices with the required rank, this is the matrix closest to Y_f in terms of the Frobenius norm. We will now construct a matrix F_s, which leads to this approximation, i.e. $Y_s = XF_s$. Define $F_s = F_f V I_s V^\top$. We have $XF_s = XF_f V I_s V^\top = Y_f V I_s V^\top = U\Lambda V^\top V I_s V^\top = U\Lambda I_s V^\top = Y_s$ as required. This can be interpreted as a two-stage process because we first find the full-rank approximation and then perform the SVD.

20.2 Weighted reduced rank regression

We are now concerned with the optimization problem as given below.

$$\begin{aligned}
&\text{argmin}_F \|XF-Y\|_\Xi = \\
&\text{argmin}_F \,\text{trace}\left((XF-Y)^\top \Xi (XF-Y)\right) = \\
&\text{argmin}_F \,\text{trace}\left((\Xi^{1/2}XF-\Xi^{1/2}Y)^\top (\Xi^{1/2}XF-\Xi^{1/2}Y)\right)
\end{aligned}$$

We see from the above that we can solve the weighted regression problem $\text{argmin}_F \|XF-Y\|_\Xi$ by computing the solution of $\text{argmin}_F \|\Xi^{1/2}XF-\Xi^{1/2}Y\|$.

21 A conjecture concerning the oblique projection matrix

To show our result concerning the pseudo-inverse, we will need a property of the oblique projection matrix that is probably true, but which we have been unable to prove. The property is as follows. Define the matrices $P = X(Y^\top X)^\dagger Y^\top$ and $P_C = XC(Y^\top XC)^\dagger Y^\top$. Assume $\mathcal{C}(X) = \mathcal{C}(XC))$. The desired property is that $P = P_C$. In the case where the matrix P_C is regarded as an oblique projection matrix, the interpretation is that the result of a projection only depends on the subspace along which we project and the subspace orthogonal to which we project, not on the choice of basis. Notice also that the property implies, if we also assume $\mathcal{C}(Y) = \mathcal{C}(YC)$, that $P = XC(C^\top Y^\top XC)^\dagger C^\top Y^\top$, by applying the original property to $P_C^\top$.

Bibliography

[1] András Antos, Csaba Szepesvári, and Rémi Munos. Learning near-optimal policies with bellman-residual minimization based fitted policy iteration and a single sample path. *Machine Learning*.

[2] Bernardo Ávila Pires. Statistical analysis of l1-penalized linear estimation with applications. Master's thesis, Univeristy of Alberta., 2011.

[3] D. Bertsekas. Temporal difference methods for general projected equations. *Automatic Control, IEEE Transactions on*, (99):1–1, 2011.

[4] Dimitri P. Bertsekas. *Dynamic Programming and Optimal Control*, chapter Approximate Dynamic Programming. 2011.

[5] Byron Boots and Geoff Gordon. Predictive state temporal difference learning. In J. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R.S. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems 23*, pages 271–279. 2010.

[6] Steven J. Bradtke and Andrew G. Barto. Linear least-squares algorithms for temporal difference learning. *Machine Learning*, 22:33–57, 1996. 10.1007/BF00114723.

[7] Kamil Ciosek. Generalizing LSTD(λ) to LSTD(λ_t). Internal UCL note., 2012.

[8] Kamil Ciosek. Option Models with Linear Features. Internal UCL note., 2012.

[9] P. T. Davies and M. K-S. Tso. Procedures for reduced-rank regression. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 31(3):pp. 244–255, 1982.

[10] Yaakov Engel. *Algorithms and Representations for Reinforcement Learning*. PhD thesis, Hebrew University, 2005.

[11] Matthieu Geist and Bruno Scherrer. l_1-penalized projected Bellman residual. In *European Wrokshop on Reinforcement Learning (EWRL 11)*, Athens, Grèce, 2011.

[12] Matthieu Geist, Bruno Scherrer, Alessandro Lazaric, and Mohammad Ghavamzadeh. A Dantzig Selector Approach to Temporal Difference Learning. In *International Conference on Machine Learning (ICML)*, 2012.

[13] Matthew Hoffman, Alessandro Lazaric, Mohammad Ghavamzadeh, and Rémi Munos. Regularized least squares temporal difference learning with nested l_2 and l_1; penalization. In Scott Sanner and Marcus Hutter, editors, *Recent Advances in Reinforcement Learning*, volume 7188 of *Lecture Notes in Computer Science*, pages 102–114. Springer Berlin / Heidelberg, 2012.

[14] J. Zico Kolter and Andrew Y. Ng. Regularization and feature selection in least-squares temporal difference learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 521–528, New York, NY, USA, 2009. ACM.

[15] Guy Lever. Private communication.

[16] O.A. Maillard, R. Munos, A. Lazaric, and M. Ghavamzadeh. Finite-sample analysis of bellman residual minimization. In *Proceedings of 2nd Asian Conference on Machine Learning (ACML2010)*, Tokyo, Japan, November 2010.

[17] C. Painter-Wakefield and R. Parr. Greedy algorithms for sparse reinforcement learning. *Arxiv preprint arXiv:1206.6485*, 2012.

[18] Ronald Parr, Lihong Li, Gavin Taylor, Christopher Painter-Wakefield, and Michael L. Littman. An analysis of linear models, linear value-function approximation, and feature selection for reinforcement learning. In *Proceedings of the 25th international conference on Machine learning*, ICML '08, pages 752–759, New York, NY, USA, 2008. ACM.

[19] Bruno Scherrer. Should one compute the Temporal Difference fix point or minimize the Bellman Residual? The unified oblique projection view. In *27th International Conference on Machine Learning - ICML 2010*, Haïfa, Israël, 2010.

[20] R. Schoknecht. Optimality of reinforcement learning algorithms with linear function approximation. In *Proceedings of the 15th Neural Information Processing Systems conference*, pages 1555–1562, 2002.

[21] David Silver. Private communication.

[22] Yi Sun, Faustino Gomez, Mark Ring, and Jürgen Schmidhuber. Incremental basis construction from temporal difference error. In Lise Getoor and Tobias Scheffer, editors, *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, ICML '11, pages 481–488, New York, NY, USA, June 2011. ACM.

[23] Richard S. Sutton and Andrew G. Barto. *Introduction to Reinforcement Learning*. MIT Press, Cambridge, MA, USA, 1st edition, 1998.

[24] J.M. Wooldridge. *Econometric Analysis of Cross Section and Panel Data*. The MIT Press, 2008.

[25] Huizhen Yu and D.P. Bertsekas. New error bounds for approximations from projected linear equations. In *Communication, Control, and Computing, 2008 46th Annual Allerton Conference on*, pages 1116 –1123, Sept. 2008.